XtreemOS

Surbhi Chitre

IRISA, Rennes, France

July 17, 2009

XtreemOS Partners

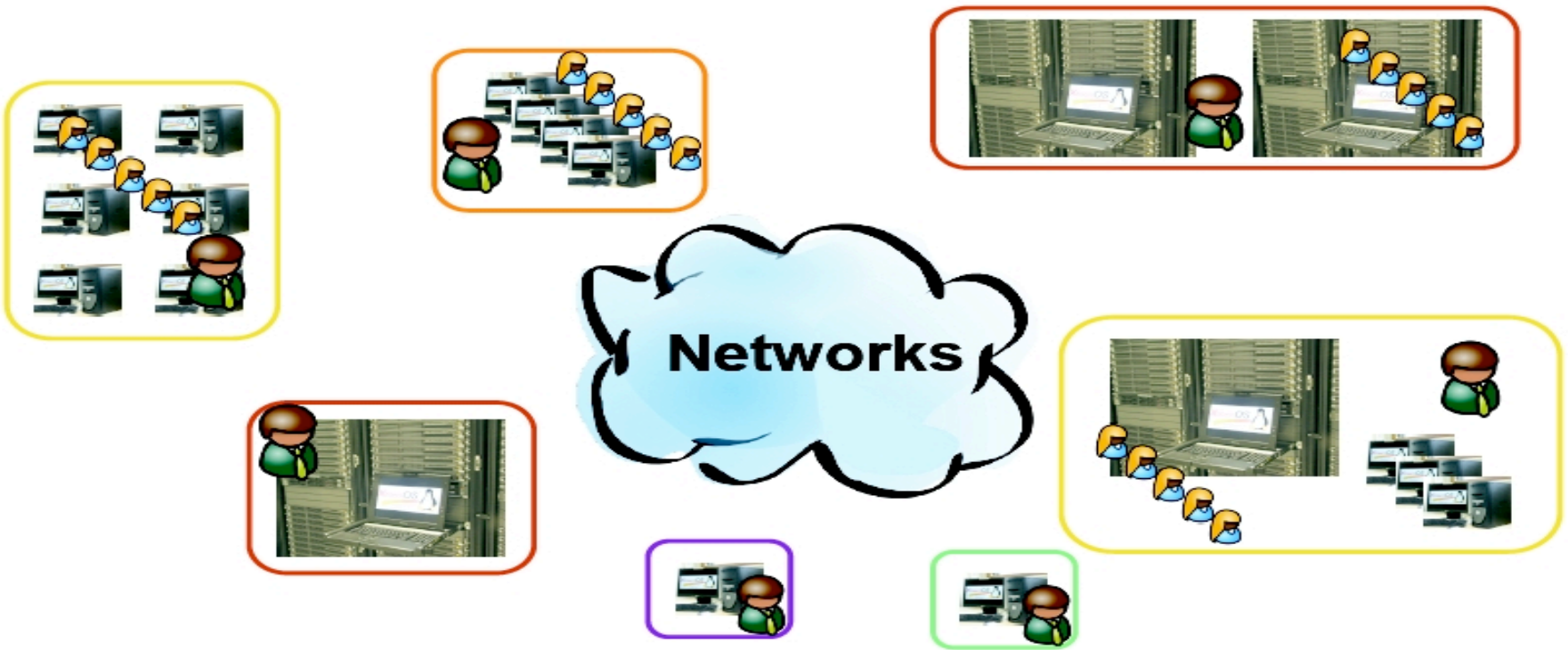
Outline

- What is XtreemOS ?
- Some features that it provides
- Job Management in XtreemOS
- OpenVZ integration in XtreemOS
- Conclusion

Discussion Path

Discussion Path

Grids



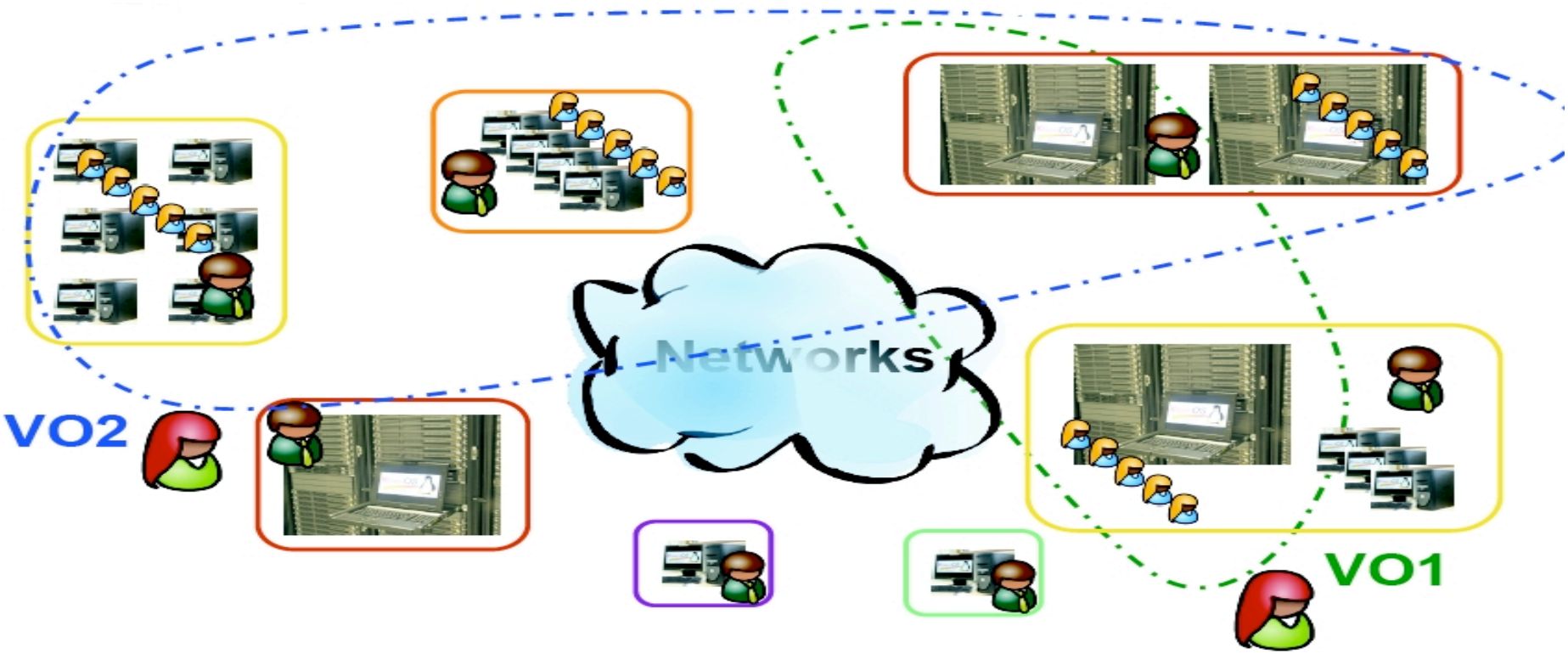
What is XtreemOS ?

- Linux based Operating System for the next generation grids
- Installation CD - mobile, PC, cluster flavor
- Distributed Resource Abstraction
- Secure Resource Sharing
- Scalable - encorporates millions of nodes, users
- High Availability - replicated services
- Legacy applications executed
- Execute applications like ./application
- Job Monitoring, isolation, fault tolerance provided

Types of Actors

1. Application Developer
   - Easy to develop applications with no modification
2. Administrator or Owner of Resource
   - Non trusted users should not be allowed
   - Node should not be attacked
3. User
   - Offload huge computations to grid
   - Security
   - Monitoring

VO Management



XtreemOS
Enabling Linux for the Grid

VO Management

Requirements

- VOs have lifespan
- Users, Resources - freely join / leave VO, members of multiple VO
- VO user account different from local account
- VO-level Policy - can a user access a VO resource
- Node Policy - can a VO user access this resource

VO Management

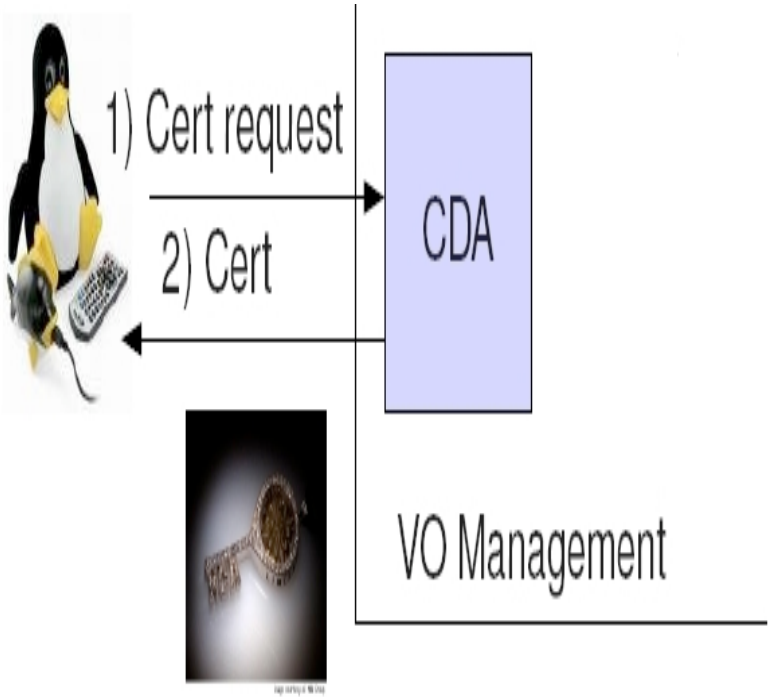
XtreemOS features

- Natively supported
- Confidentiality, Integrity, Authenticity provided
- Manages VO lifecycle, resources, users and credentials.
- Enforces VO and node policies upon resource usage

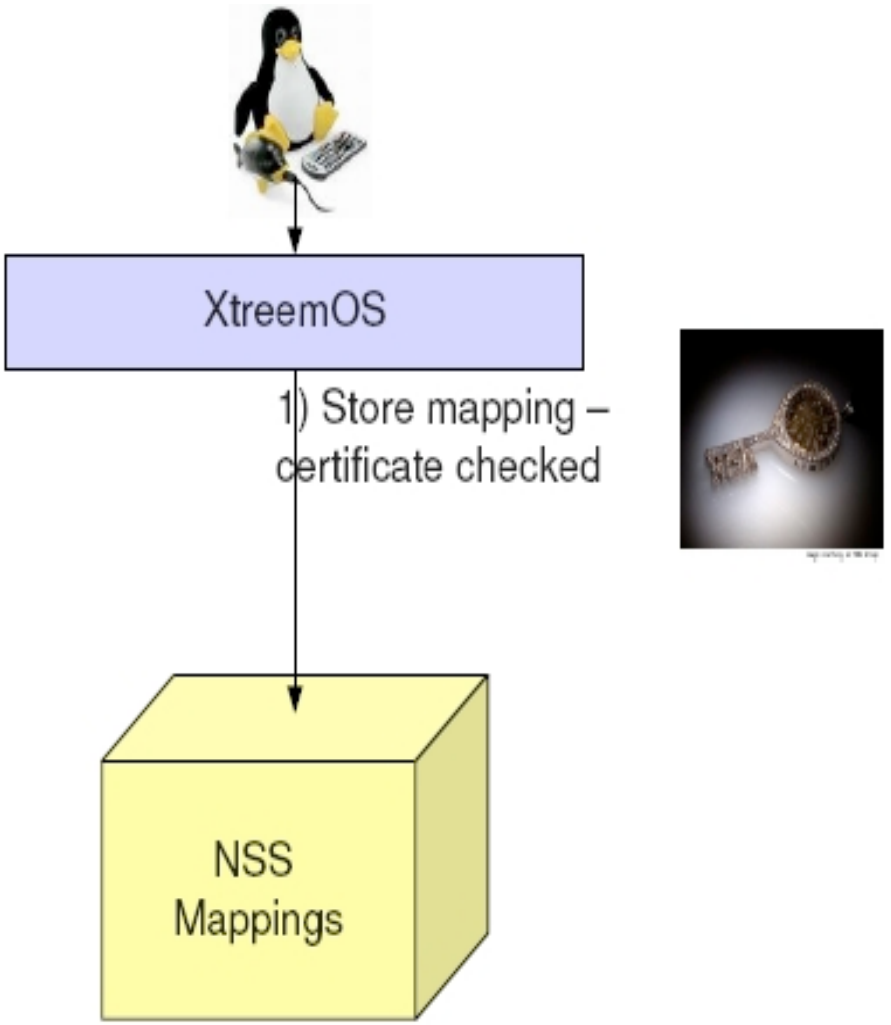
Security

- Single Sign On
- Pluggable Authentication Module
- Name service switch and key retention - Session Management
- Grid/VO users are never local users

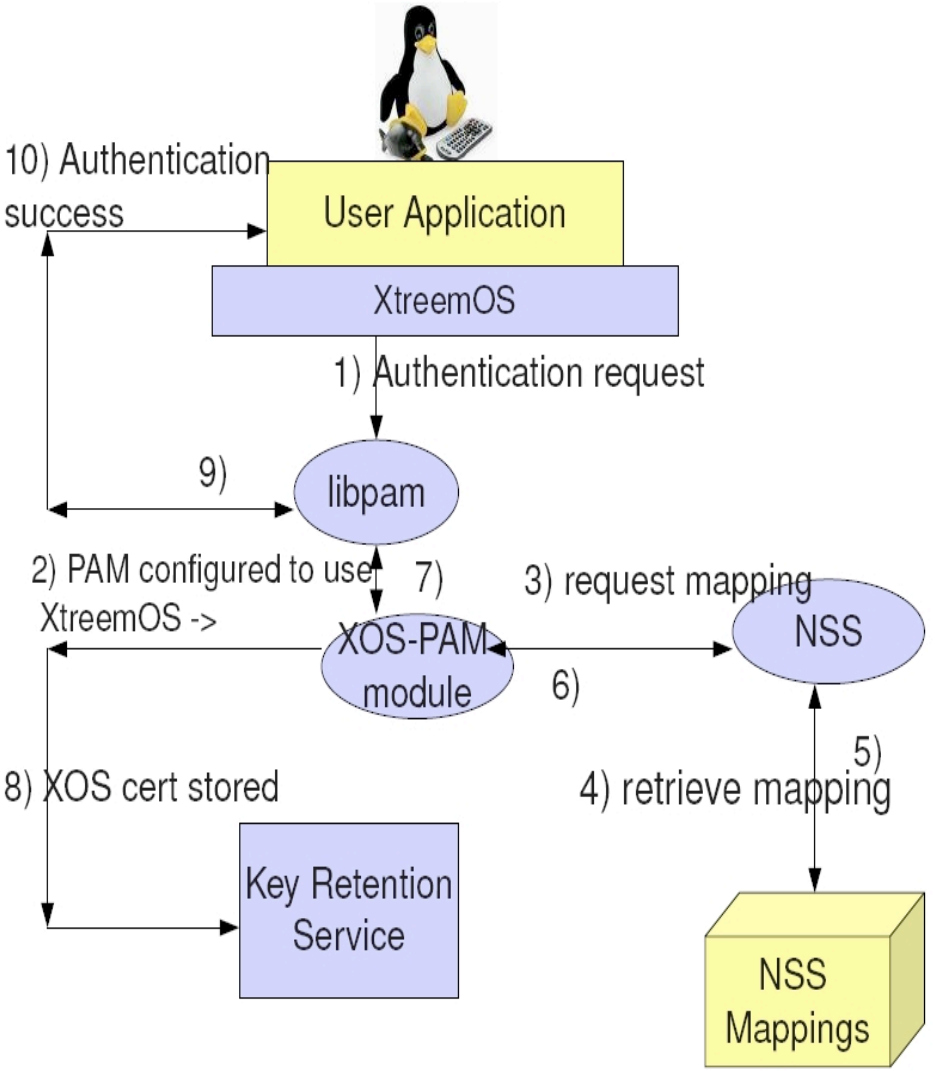
Single Sign On



Single Sign On



Single Sign On



Entities: Users, Resources and Services

- Resource status discovered - advanced p2p techniques
- Services are decentralised and replicated
- User information is stored in a DHT.

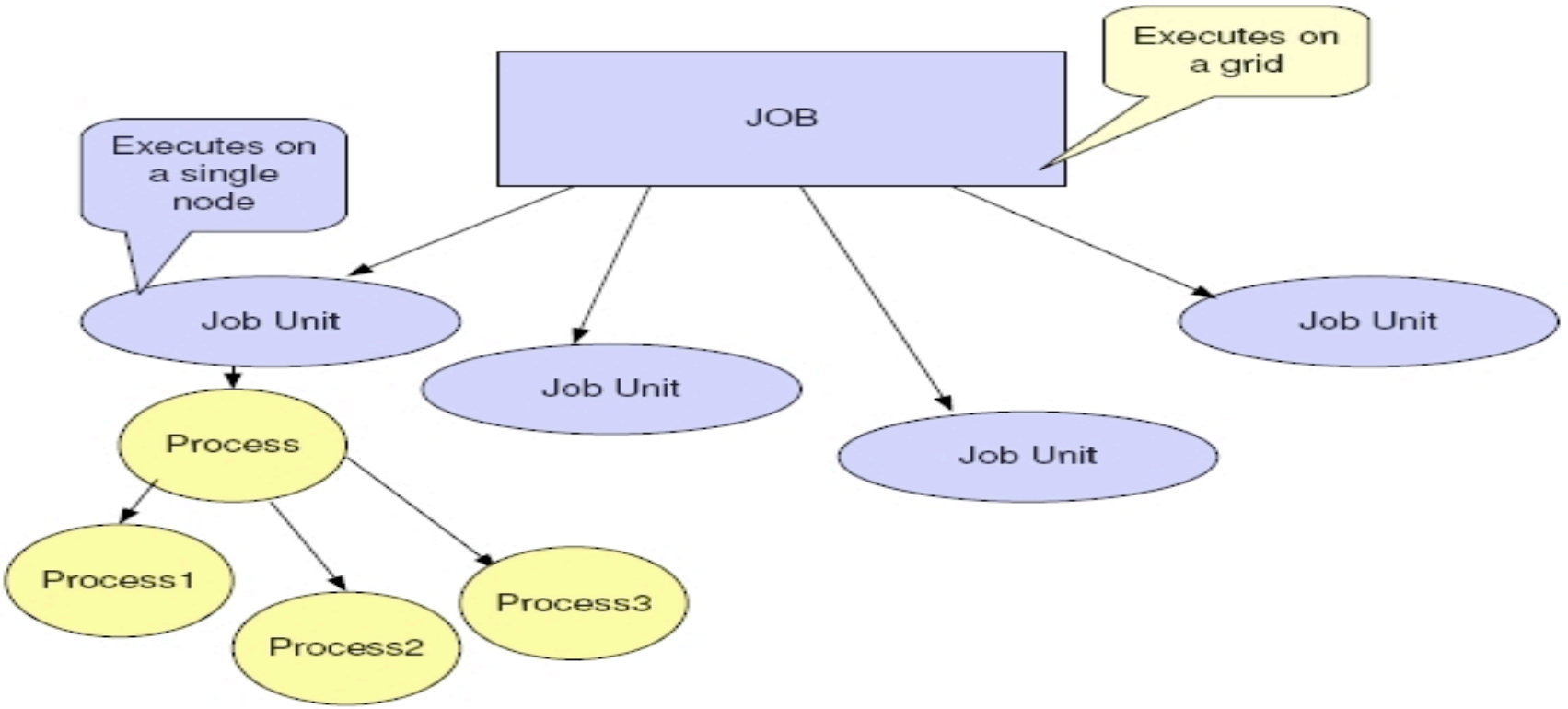
Data Management - XtreemFS

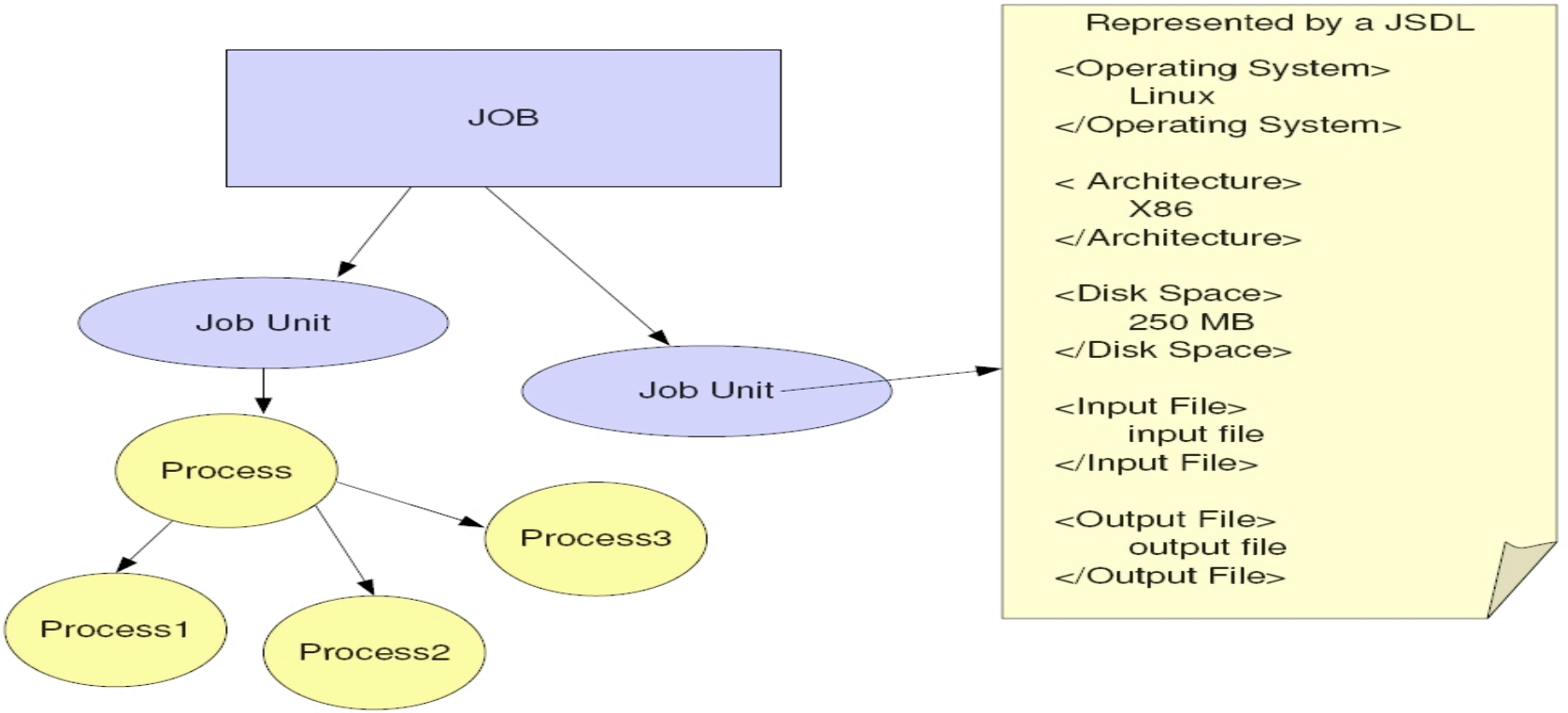
- Posix like, Distributed, spanning the grid
- Self replicating, striping support
- Transactional consistency
- Data stored on data servers
- Client - server model, fs requests translated to RPC
- Grid User has a corresponding fs space
- Automatically mounted when a user executes a job

Job Management

- Offload execution of heavy jobs in the outside grids!
- Execute jobs securely
- Have control over your jobs
- When it fails because of some problem outside the job, restart it from a last know point
- Debug a job if it fails from a last know point
- Store the output securely.

Discussion Path


JOB
Executes on a grid
Executes on a single node
Job Unit
Job Unit
Job Unit
Job Unit
Process
Process1
Process2
Process3


JOB
Job Unit
Job Unit
Process
Process3
Process1
Process2
Represented by a JSDL
<Operating System>
Linux
</Operating System>
< Architecture>
X86
</Architecture>
<Disk Space>
250 MB
</Disk Space>
<Input File>
input file
</Input File>
<Output File>
output file
</Output File>

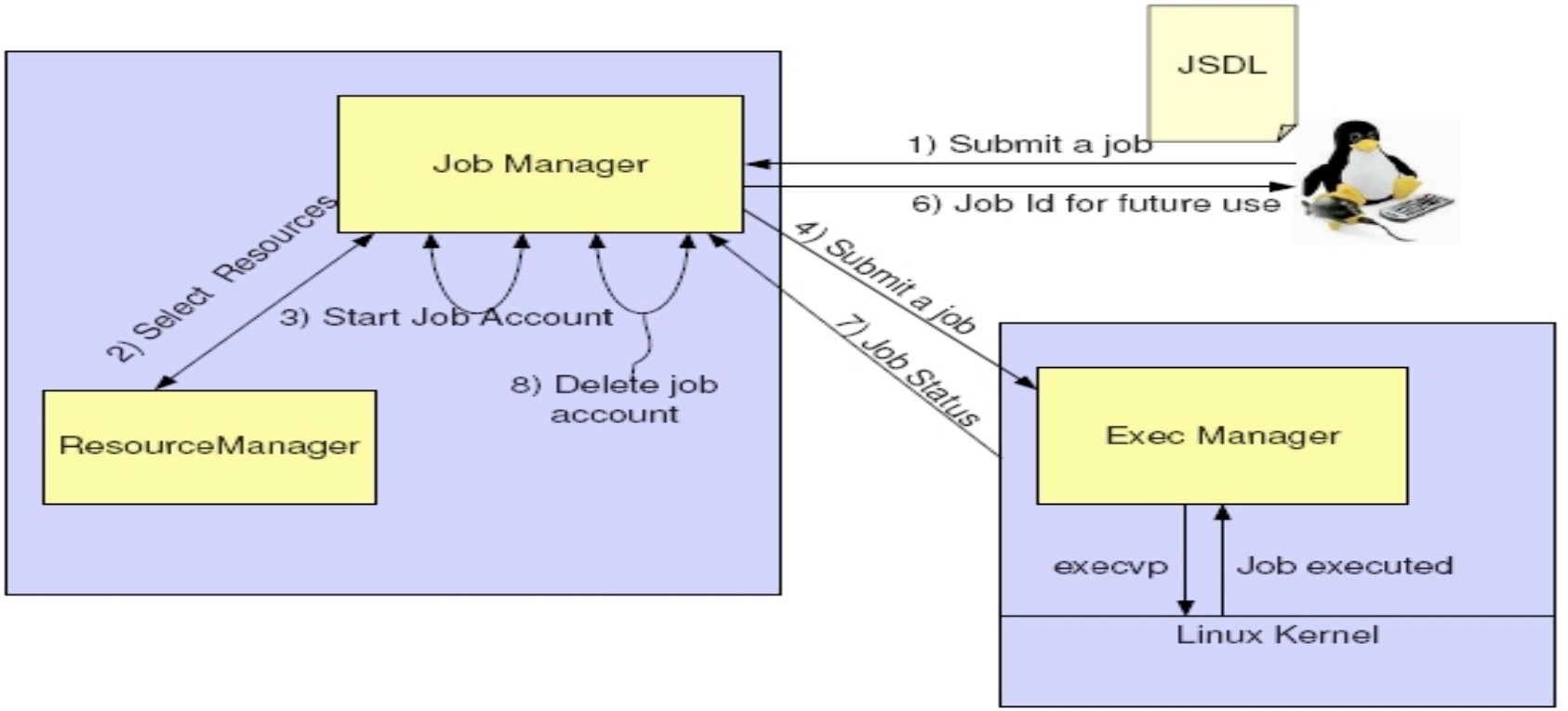
Runs all Distributed XtreemOS Services. Example:
- Runs the job managers
- Manages the jobs
- Does resource discovery and resource selection
- Stores the user information in DHT

Core Node

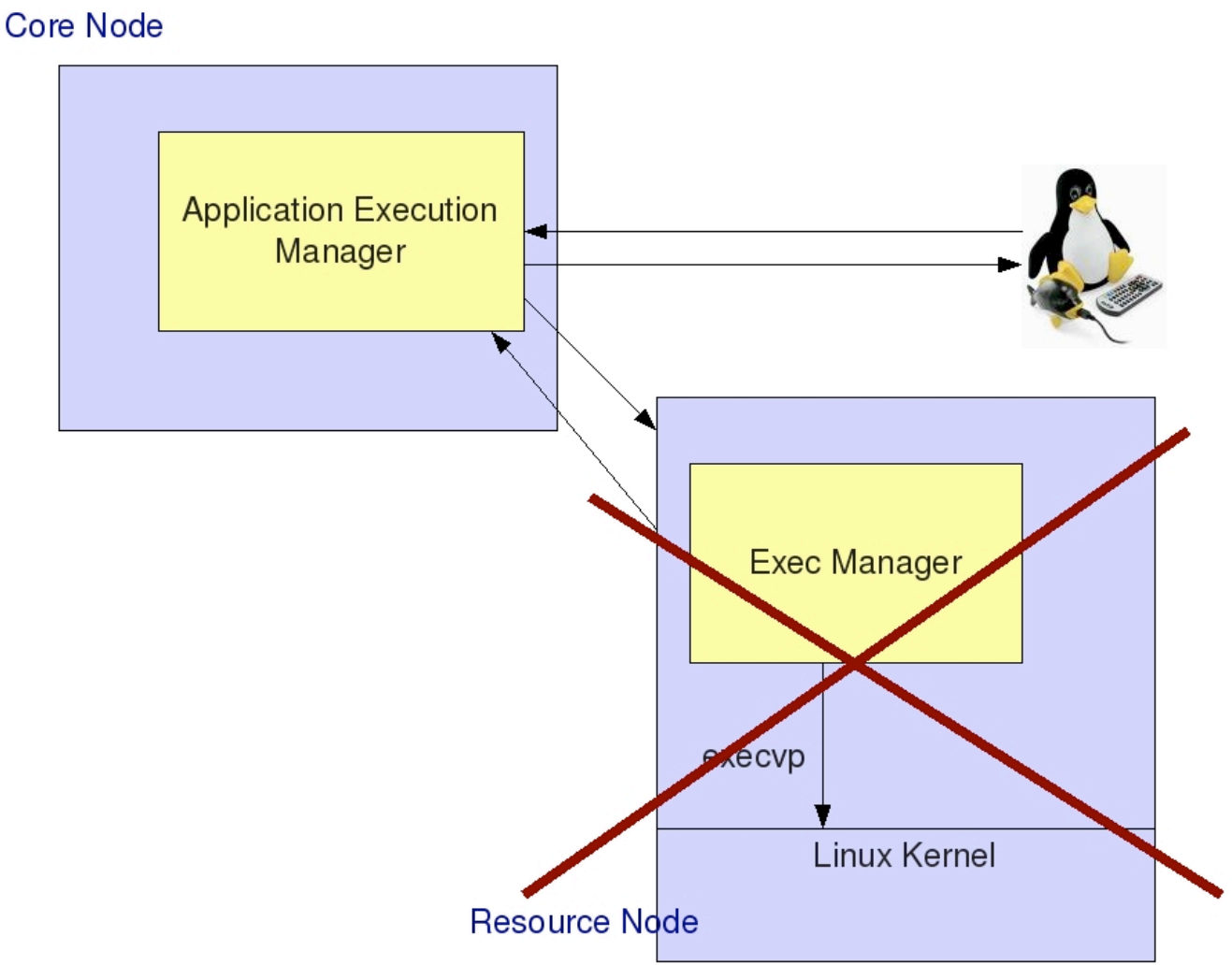
Runs all services required on a single node for a job Unit. Example:
- Runs the execution managers
- Executes the jobs
- Authenticates the users as per VO policy and the node policy

Resource Node

XtreemOS
Enabling Linux for the Grid


JSDL
Job Manager
1) Submit a job
6) Job Id for future use
2) Select Resources
3) Start Job Account
8) Delete job account
4) Submit a job
7) Job Status
ResourceManager
Exec Manager
execvp
Job executed
Linux Kernel

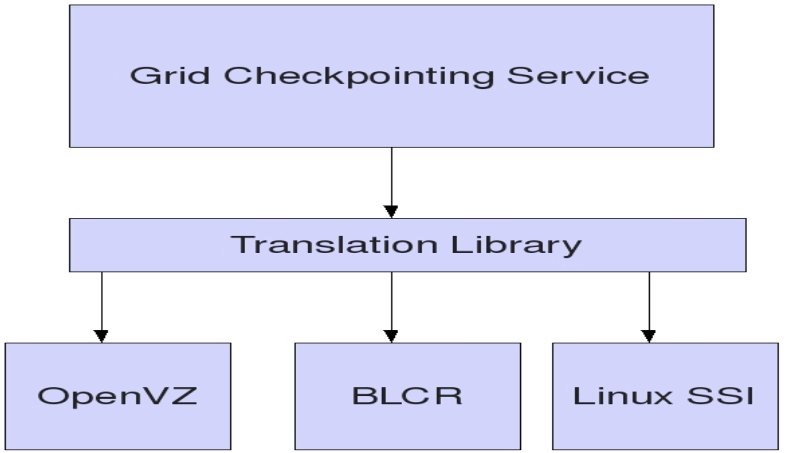
Job/Node failure



Why checkpoint

- Grid Node stops participation, restart job on some other node
- Debugging
- Recover job from some last known state
- Checkpointing - restart used

Multiple checkpointers



Discussion Path

- Operating system evolution.
- Virtual Private Server
- Containers - root access, separate filesystem, process tree, network stacks, IPC objects and other resources
- Resources - limits and gurantees.
- Containers have an id and a state
- Containers - checkpointed, restarted, migrated.
- On restart/migration - network connections can be resumed.

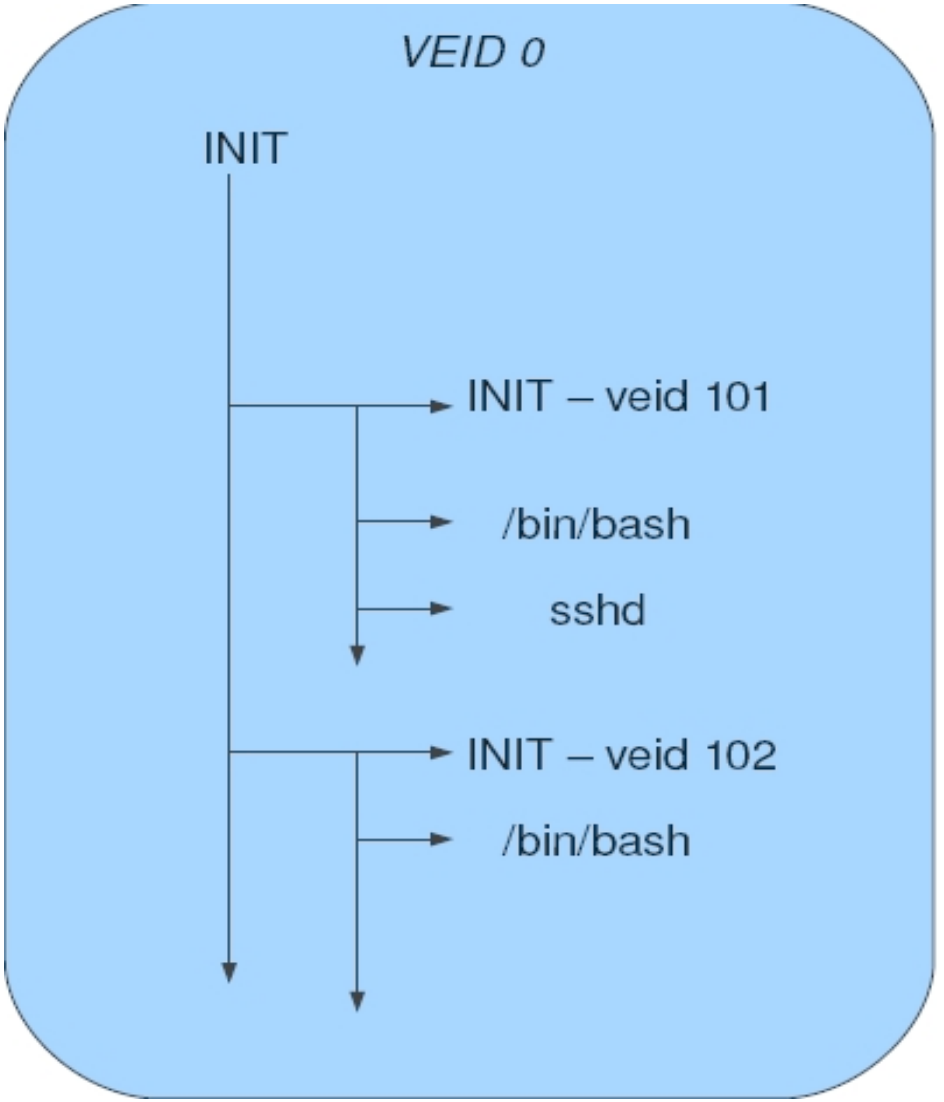
On the surface - OpenVZ kernel?

- Root container - id always 0
- *All* the processes now start in a root container
- A new container is a child of this container.

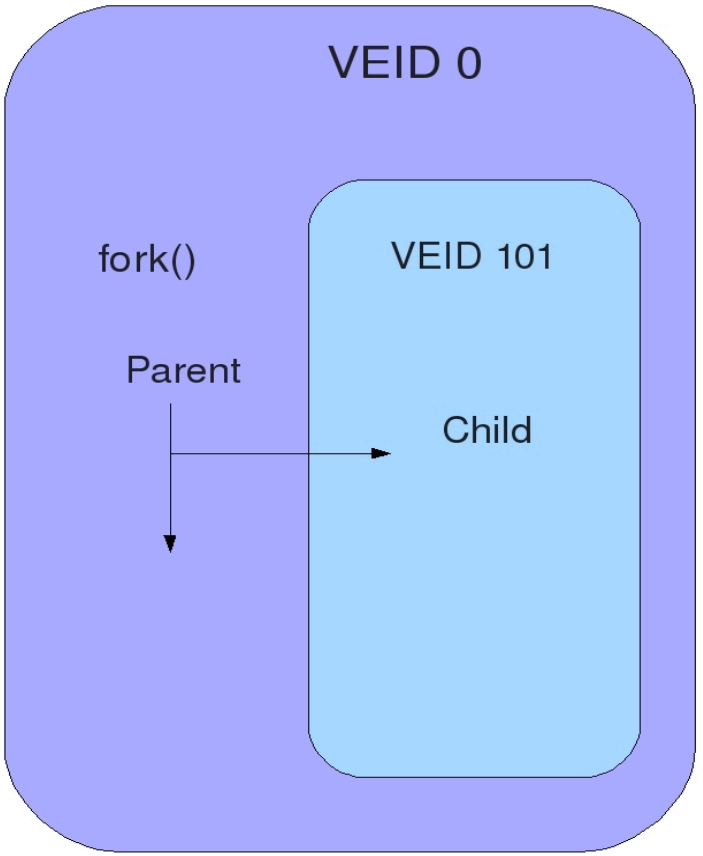
What happens when you start a container?

- A new VPS created.
- Virtual process id and real process id
- Processes outside the container - real process id.
- Processes inside the container - virtual process id.
- Processes in the container can be seen from root container (ps/pstree)

process hierarchy



Foreign dependencies - Cannot checkpoint



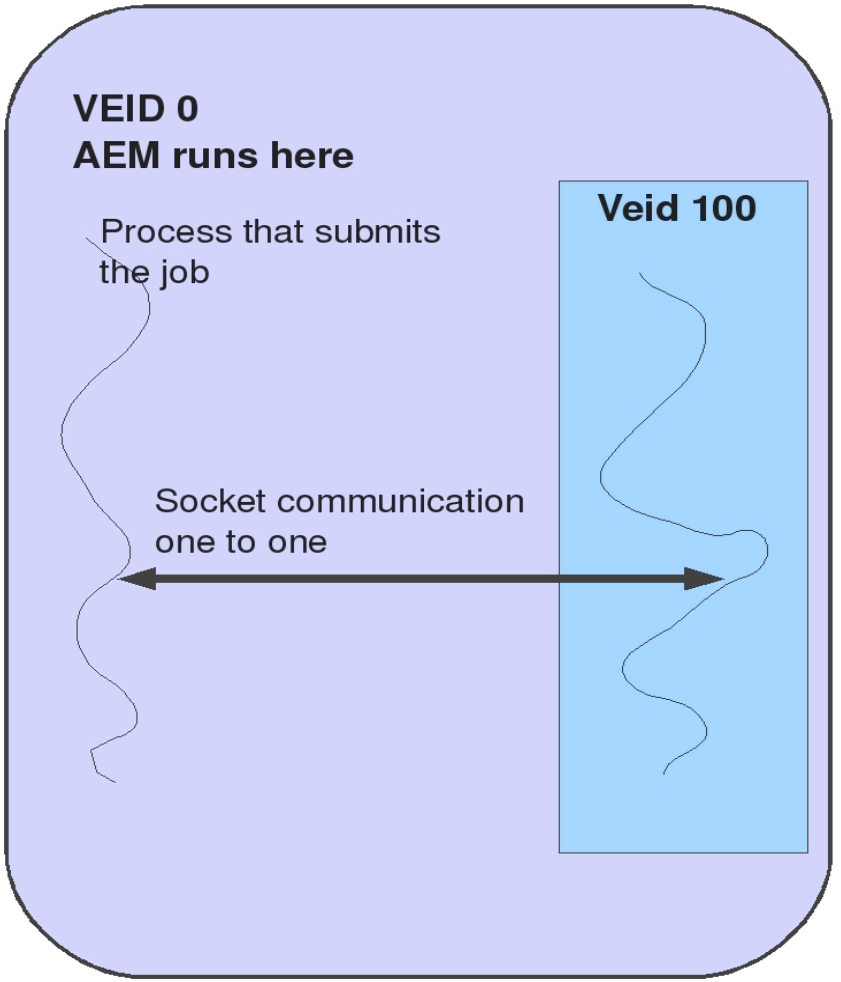
Requirement

- Submit a job to a container through the Application execution manager of XtreemOS.
- Track a job execution
- Checkpoint, restart and migrate a job through the Checkpoint Restart Manager of XtreemOS

Job Submission

- Job should be submitted to a container than to a native kernel.
- Identify that the job needs OpenVZ - JSDL
- A new container should be created - needs root access
- XtreemFS dir should be accessed by the corresponding user.
- Job submission with no foreign process dependency - does not allow checkpointing.

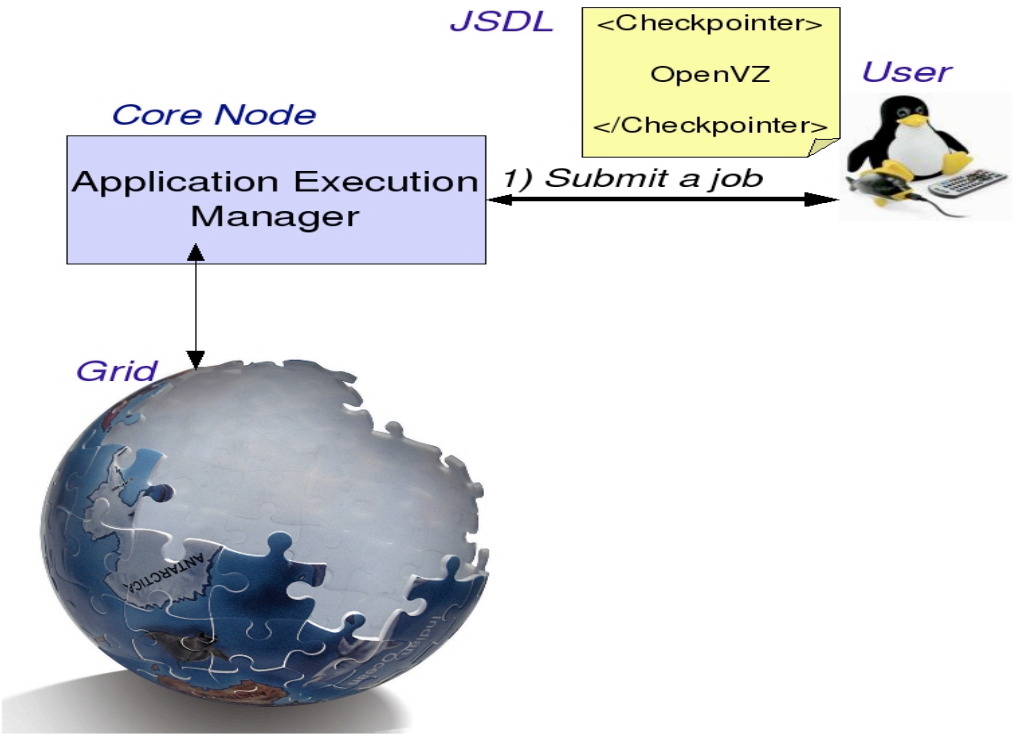
Simple solution - socket communication



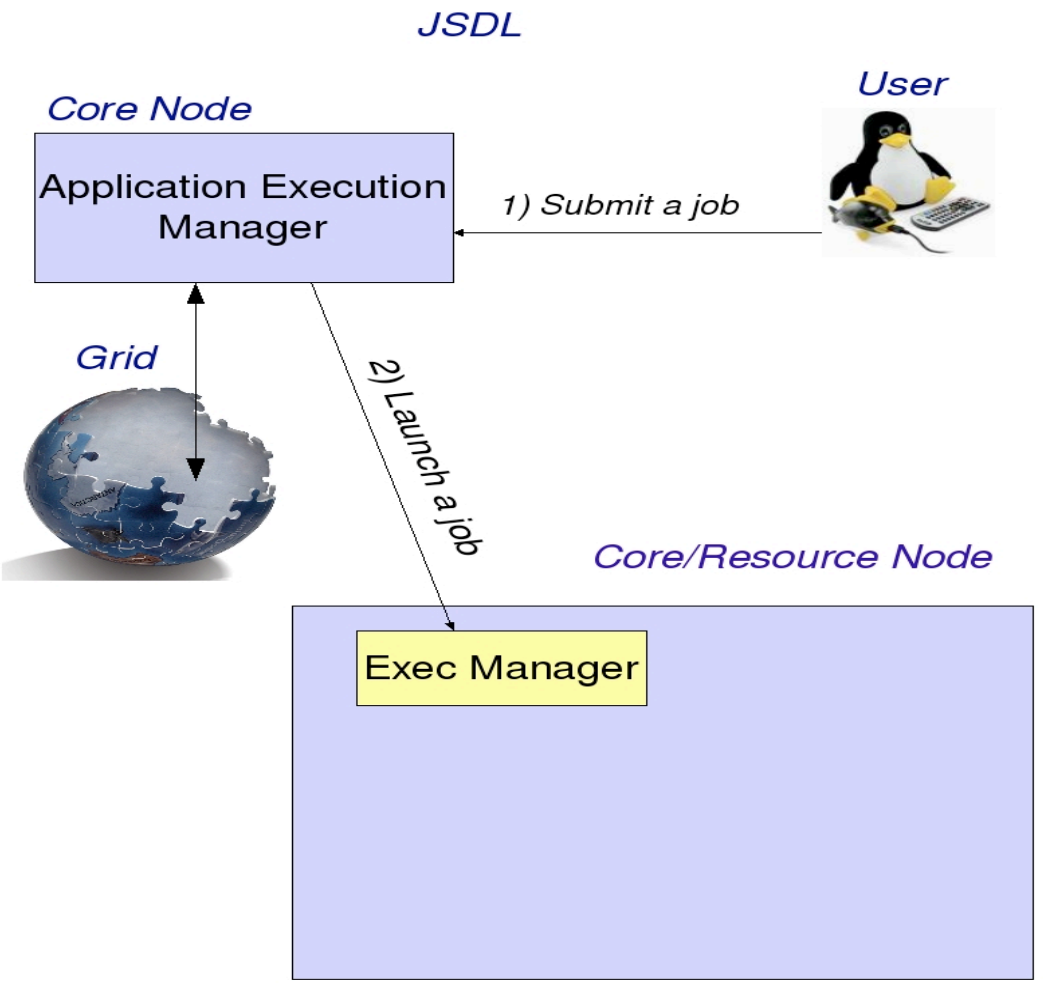
Job Submission

- loader application launches a grid job.
- loader calls setgid and setuid appropriately.
- loader application helps in job tracking

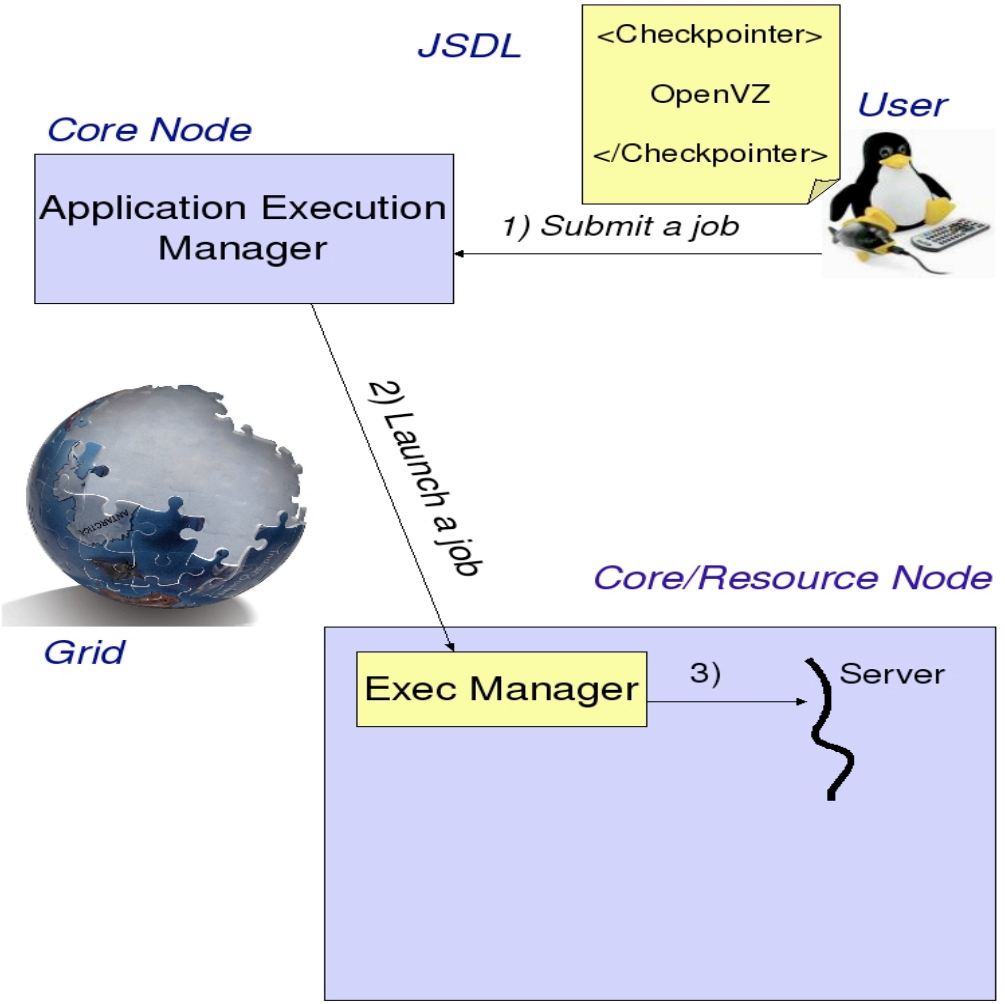
Job submission



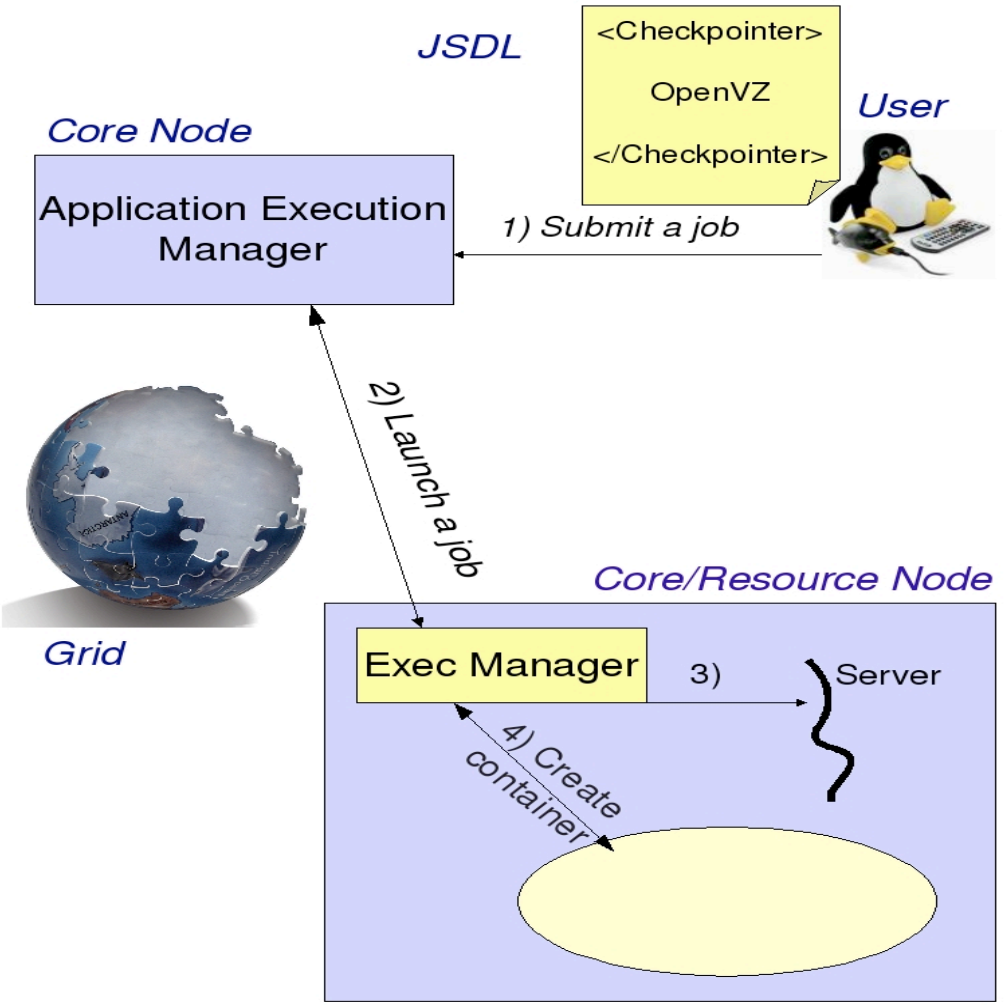
Job submission



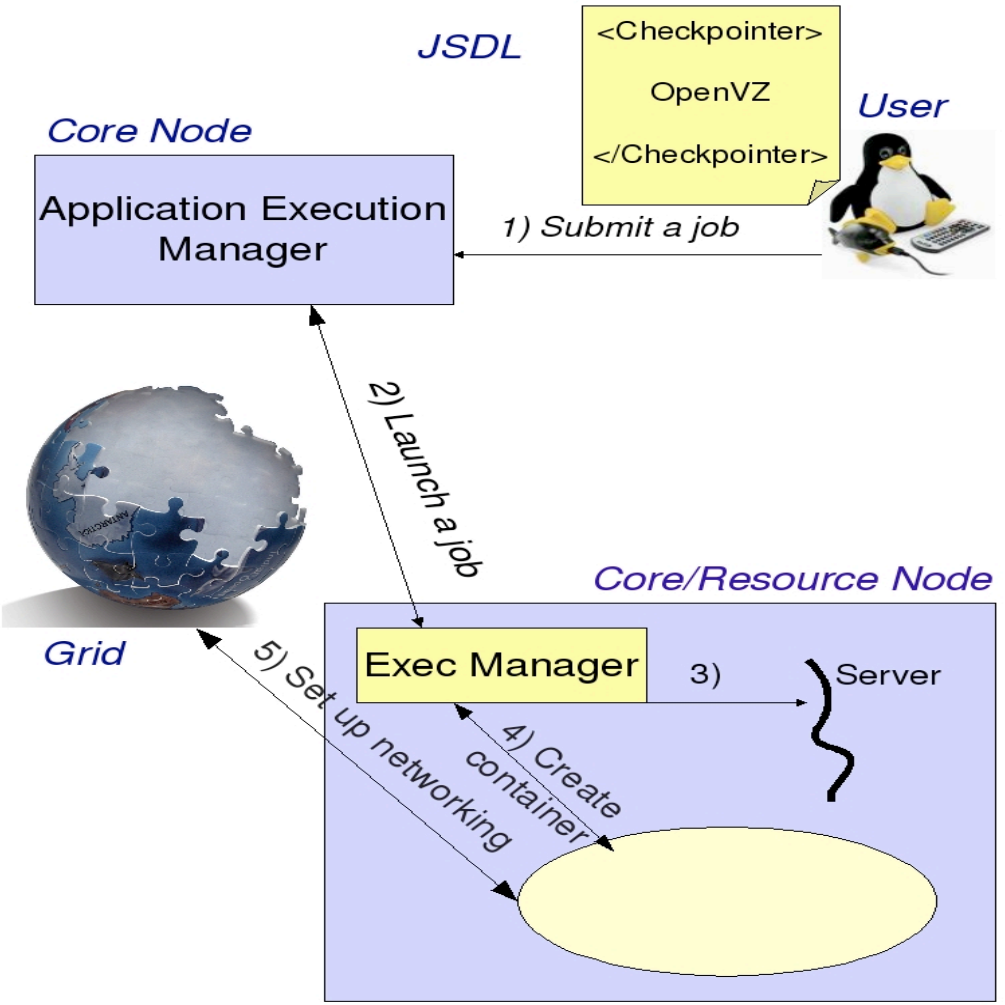
Job submission



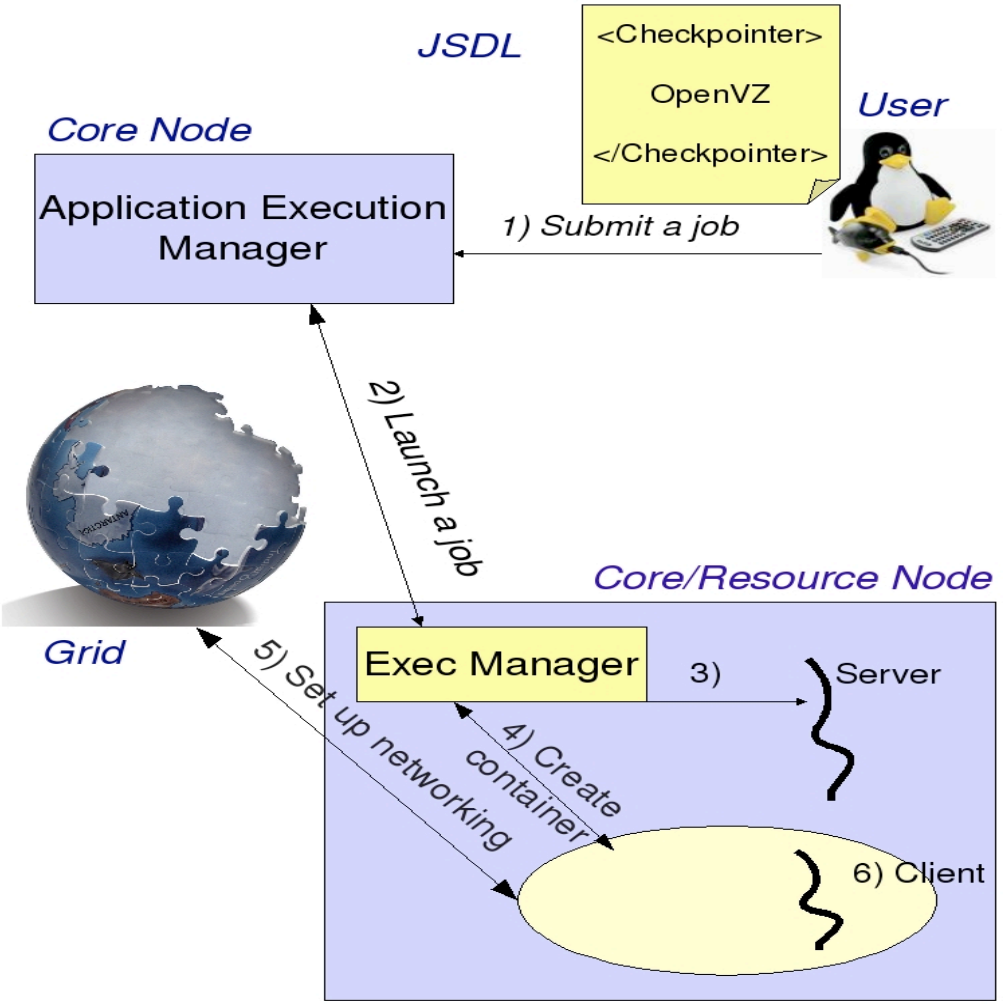
Job submission



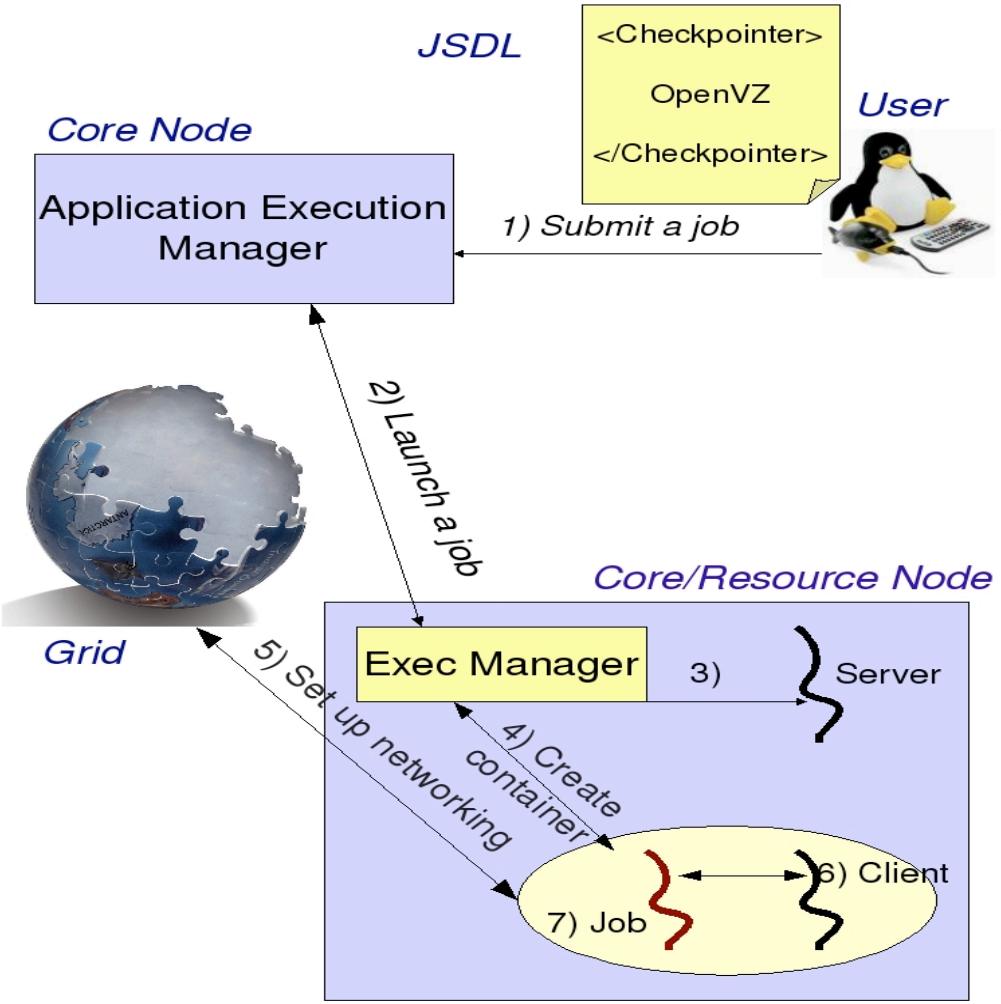
Job submission



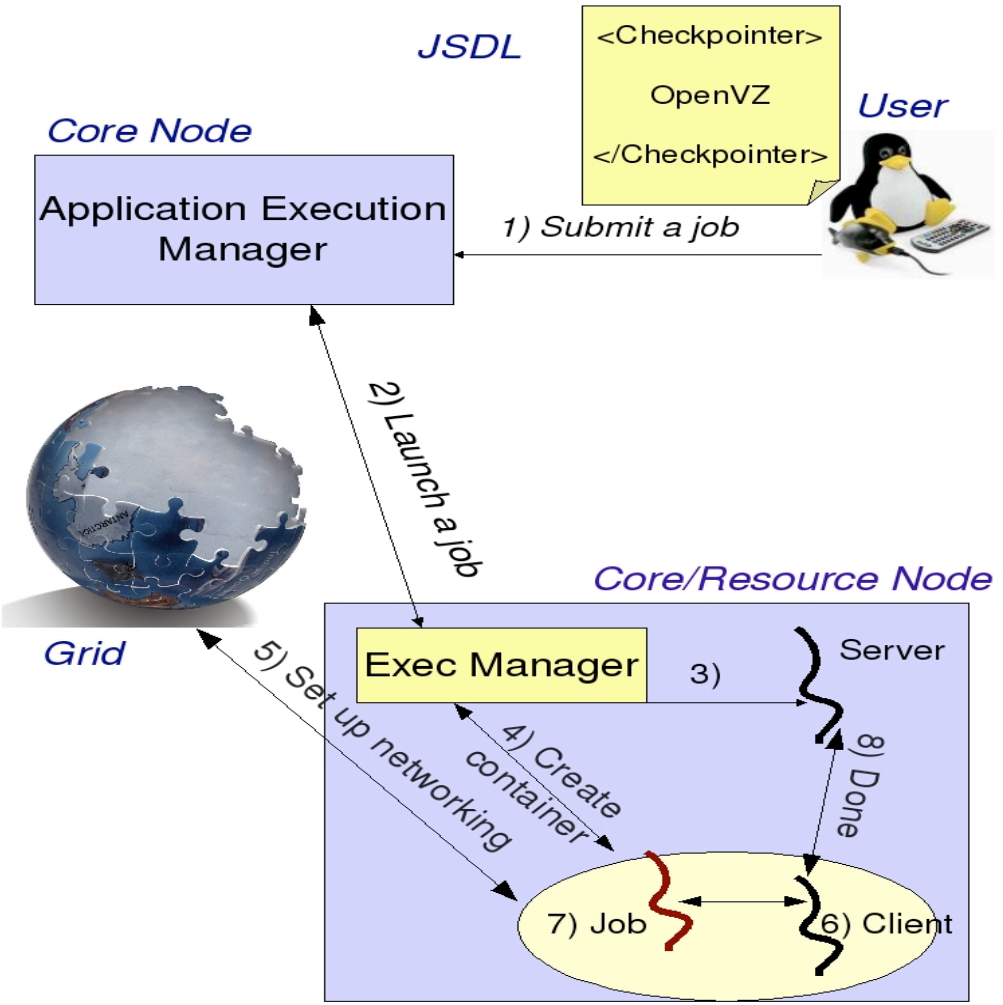
Job submission



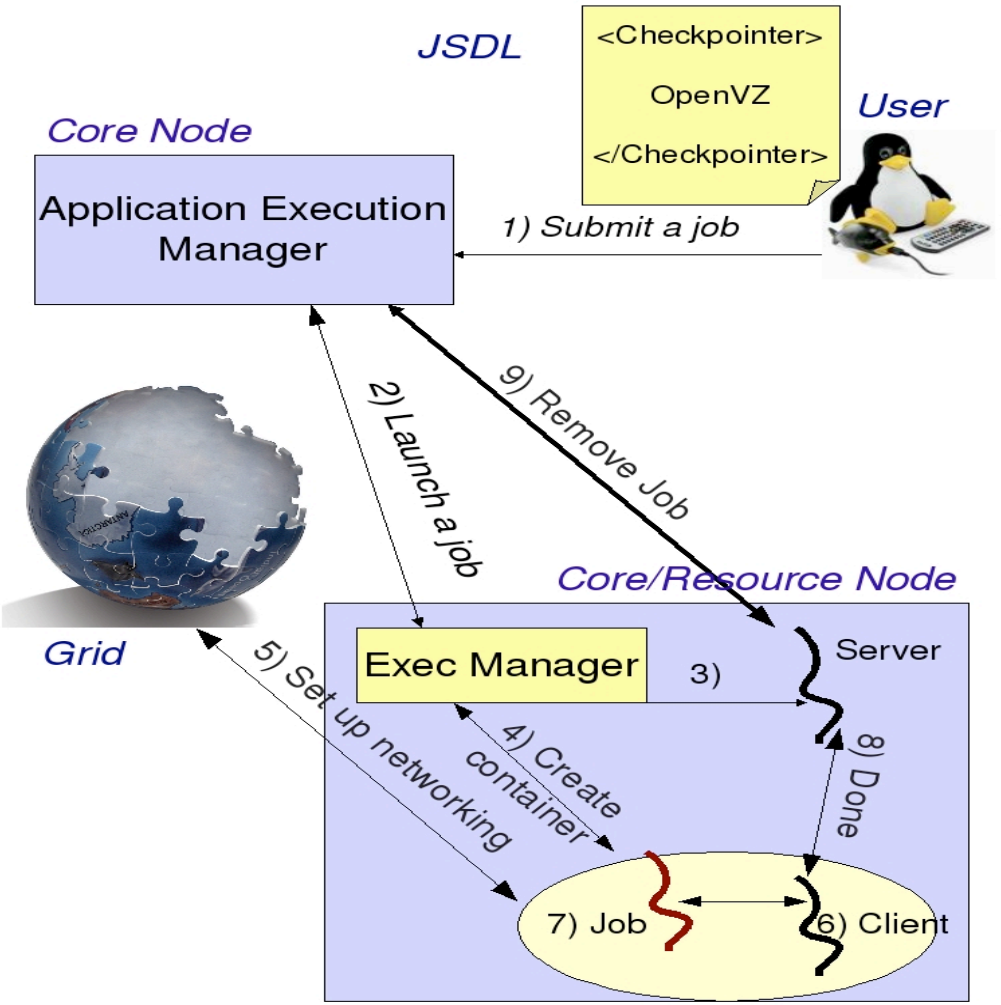
Job submission



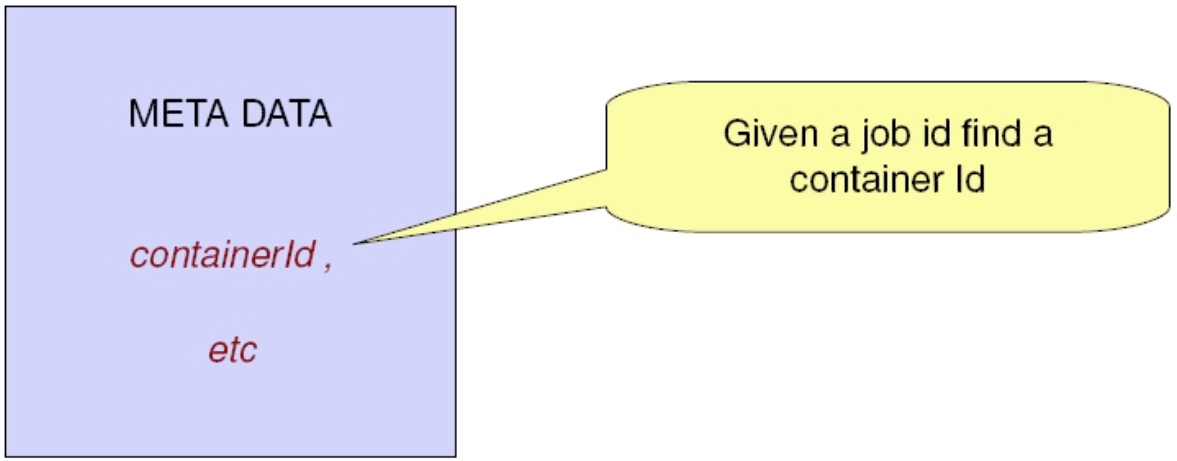
Job submission



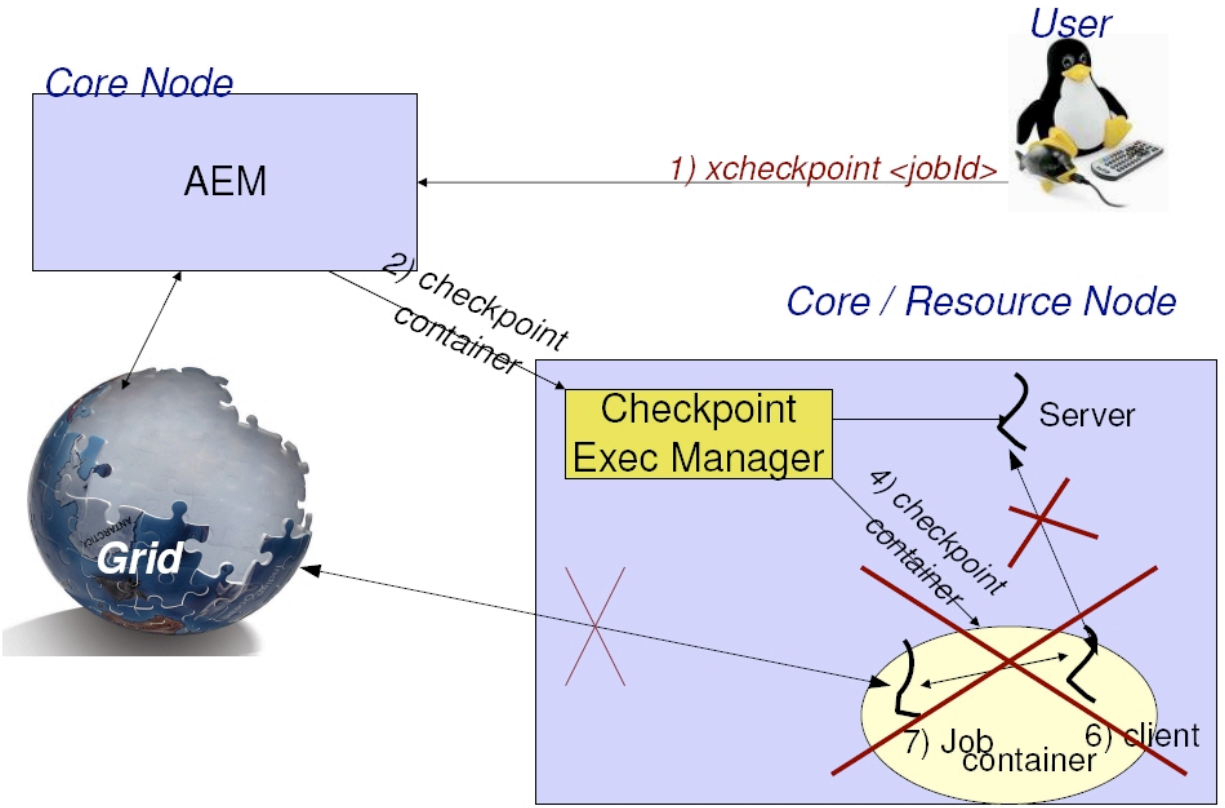
Job submission



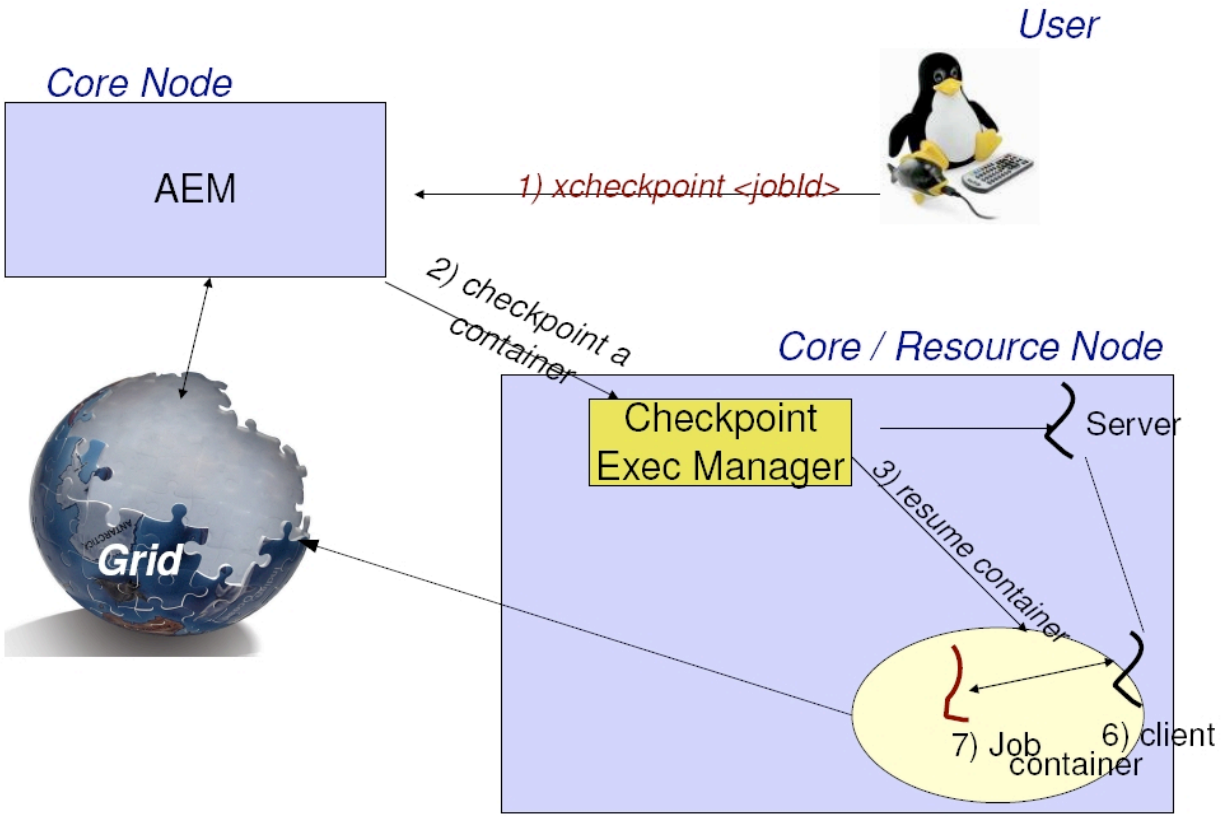
MetaData - Checkpoint



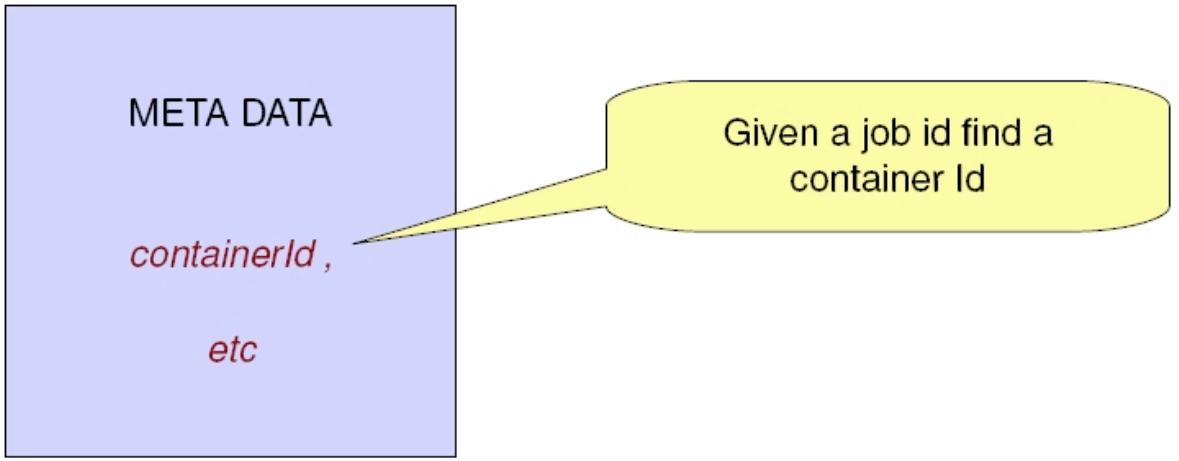
Checkpoint


User
Core Node
AEM
1) xcheckpoint <jobId>
2) checkpoint container
Core / Resource Node
Checkpoint Exec Manager
Server
4) checkpoint container
Grid
7) Job container
6) client

Checkpoint-Resume

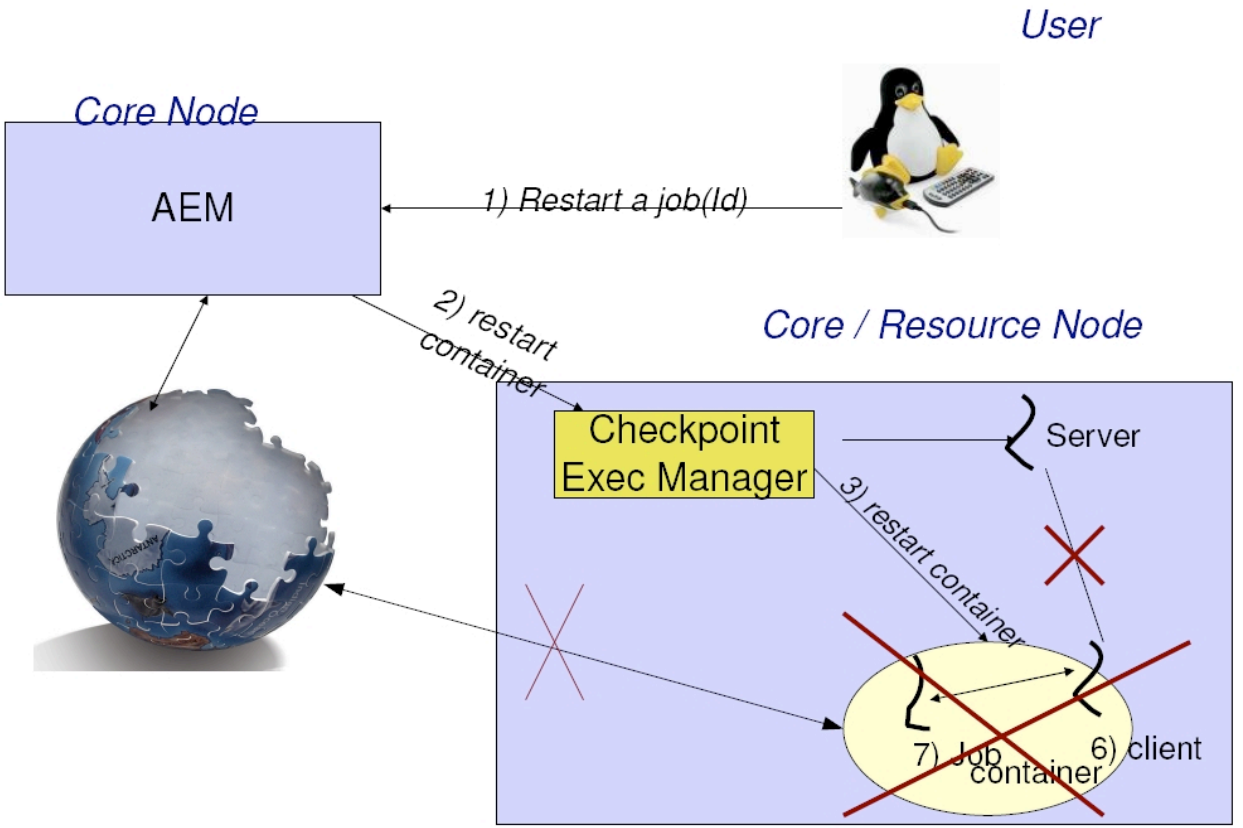


MetaData - Restart



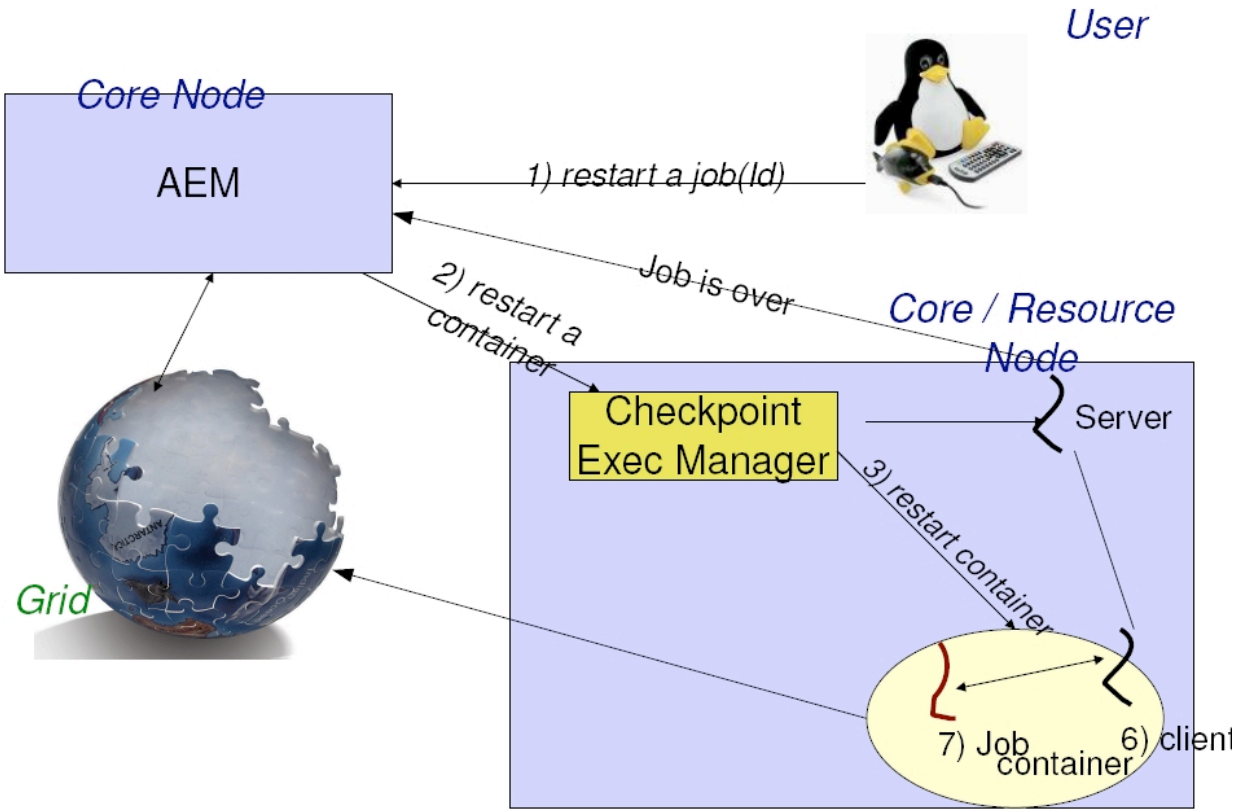
Also Identify where the dump file is stored for a particular job

Restart



XtreemOS
Enabling Linux for the Grid

Restart


User
Core Node
AEM
1) restart a job(Id)
Job is over
2) restart a container
Core / Resource Node
Checkpoint Exec Manager
Server
3) restart container
Grid
7) Job container
6) client

To sum it

Integration

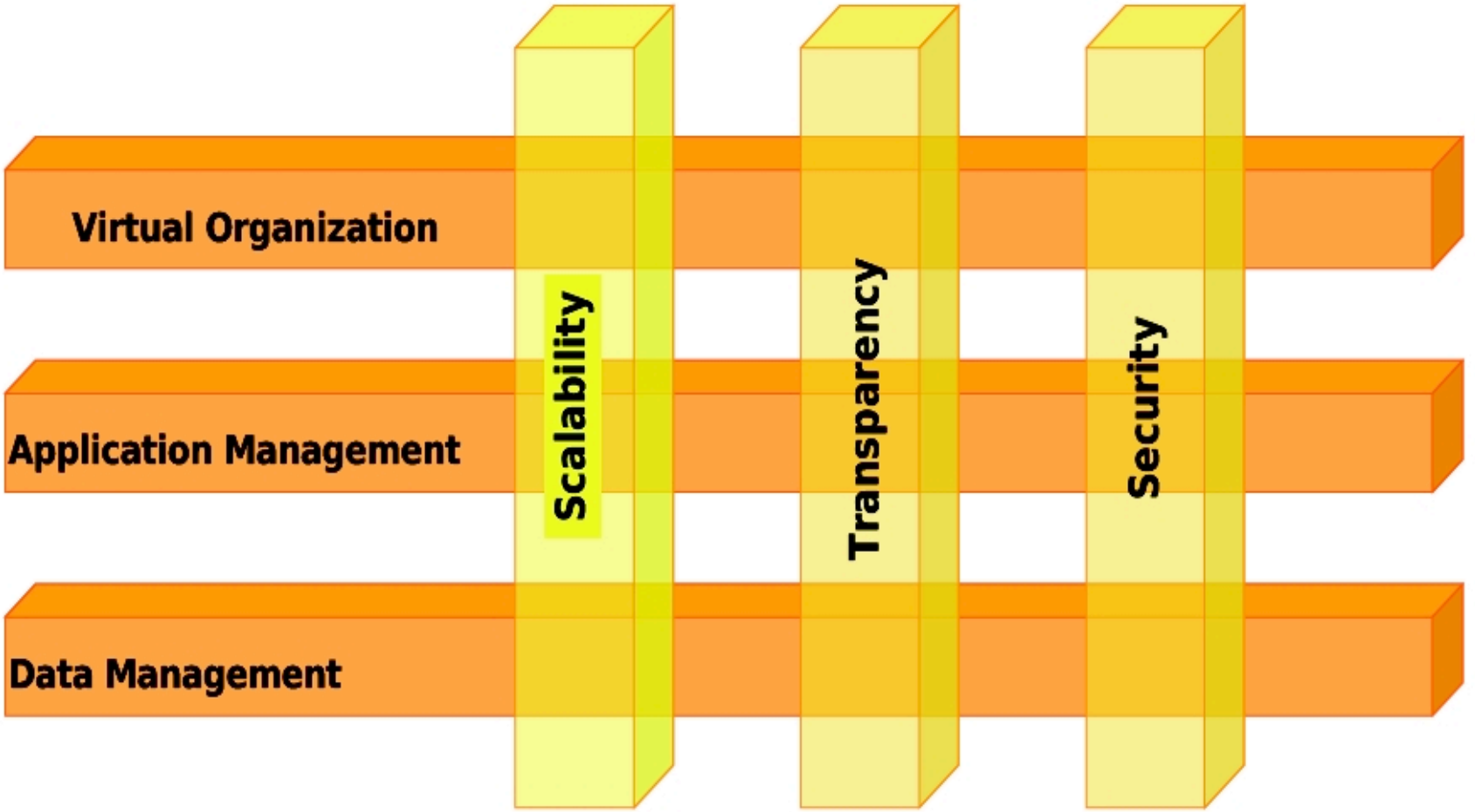
- Foreign process dependency should be avoided at job submission for enabling checkpointing.
- Job tracking is important for job monitoring and cleanup of job
- OpenVZ integration lets an XtreemOS be submitted to a container.

Discussion Path

New Features - Why XtreemOS ?

- Not a middleware but a OS distribution
- All required grid related components in one CD
- Support for interactive applications. More is coming soon!
- execute legacy application the legacy way - ./elf-executable

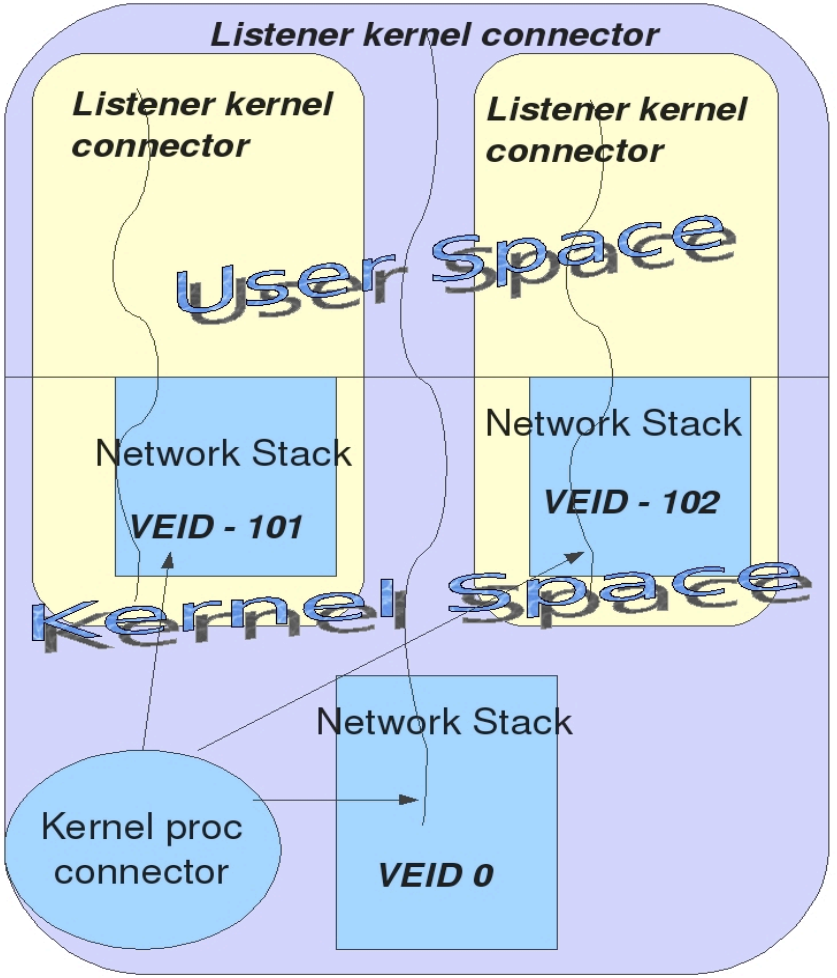
Conclusions



Thank You !

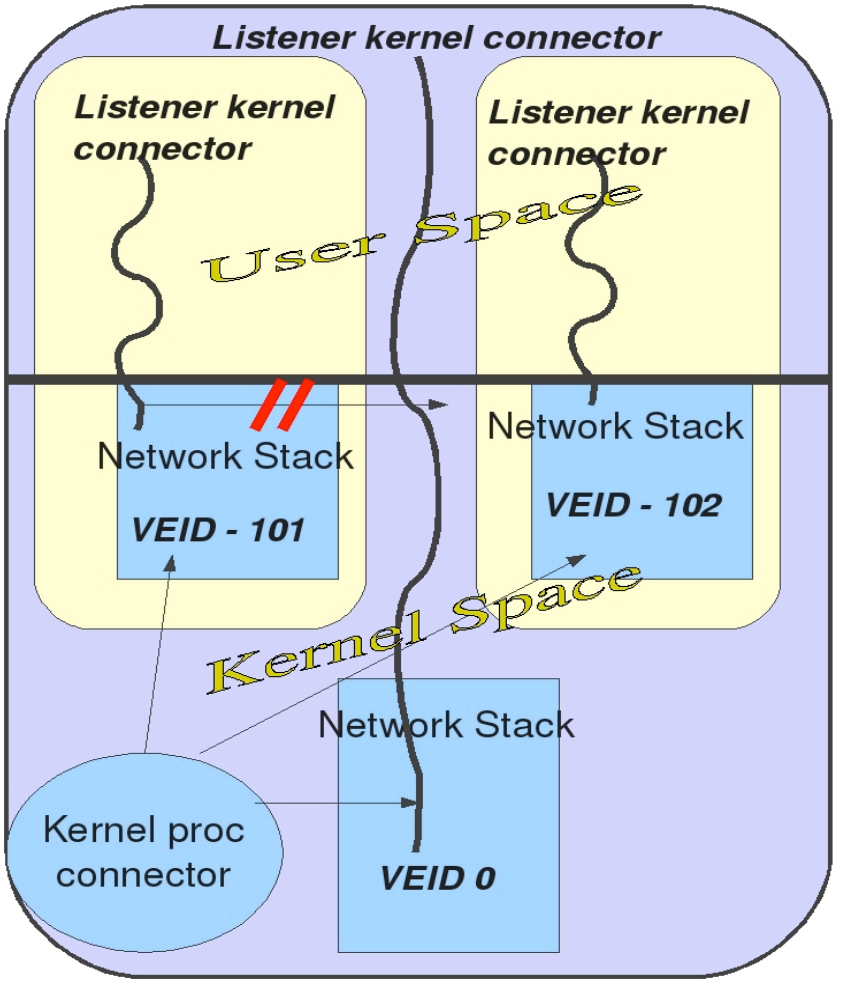
- http://xtreemos.org
- http://xtreemfs.org
- http://wiki.mandriva.com/en/Releases/Mandriva/XtreemOS

Kernel Connector



XtreemOS
Enabling Linux for the Grid

Kernel Connector bound to a container



XtreemOS
Enabling Linux for the Grid